

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (Presently Amended) A computer system that employs a plurality of execution
2 threads to perform tasks that the threads identify dynamically, the computer
3 system being so programmed as to:
 - 4 A) provide a plurality of task queues, each of which is associated with a
5 different ordered pair of the threads, one thread of the ordered pair being
6 denominated the enqueueer of that queue and the other being
7 denominated the dequeueer thereof, wherein the execution threads operate
8 so that only an enqueueer of a task queue adds entries to that task queue
9 and only the dequeueer of a task queue removes entries from that task
10 queue;
 - 11 B) when ~~one said~~ a thread identifies a task, ~~pushes~~ push an identifier of that
12 the task thus identified onto a set of at least one of the queues of which
13 that thread is an enqueueer; and
 - 14 C) when ~~one said~~ a thread requires one of the dynamically identified tasks to
15 perform, ~~causes~~ cause that thread to perform a task identified by a task
16 identifier fetched by that thread from a task queue of which that thread is
17 the dequeueer.
- 1 2. (Original) A computer system as defined in claim 1 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

- 1 3. (Original) A computer system as defined in claim 1 wherein each said task
2 identifier is an identifier of the object with which the task is associated.
- 1 4. (Original) A computer system as defined in claim 3 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.
- 1 5. (Original) A computer system as defined in claim 3 wherein each said task
2 identifier is a pointer to the object with which the task is associated.
- 1 6. (Original) A computer system as defined in claim 1 wherein, when one said
2 thread identifies a task, the computer system pushes an identifier of that thread
3 onto only one of the queues of which that thread is an enqueueer.
- 1 7. (Original) A computer system as defined in claim 6 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.
- 1 8. (Original) A computer system as defined in claim 1 wherein identifiers of tasks
2 successively identified by a given thread are not in general pushed onto the
3 same queue.

- 1 9. (Original) A computer system as defined in claim 8 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.
- 1 10. (Original) A computer system as defined in claim 1 wherein a task queue is
2 provided for each ordered pair of the threads.
- 1 11. (Original) A computer system as defined in claim 10 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.
- 1 12. (Presently Amended) For using a computer system to employ a plurality of
2 execution threads to perform tasks that the threads identify dynamically, a
3 method that includes:
- 4 A) providing a plurality of task queues, each of which is associated with a
5 different ordered pair of the threads, one thread of the ordered pair being
6 denominated the enqueueer of that queue and the other being
7 denominated the dequeueer thereof, wherein the execution threads operate
8 so that only an enqueueer of a task queue adds entries to that task queue
9 and only the dequeueer of a task queue removes entries from that task
10 queue;
- 11 B) ~~when one said a~~ thread identifies a task, pushing an identifier of that the
12 task thus identified onto a set of at least one of the queues of which that
13 thread is an enqueueer; and

14 C) when ~~one said~~ a thread requires one of the dynamically identified tasks to
15 perform, causing that thread to perform a task identified by a task identifier
16 fetched by that thread from a task queue of which that thread is the
17 dequeuer.

1 13. (Original) A method as defined in claim 12 wherein each said dynamically
2 identified task is the garbage-collection task of performing, for a given object
3 associated with that task, processing that includes identifying in the given object
4 references to other objects and thereby identifying the tasks of performing similar
5 processing for those other objects.

1 14. (Original) A method as defined in claim 12 wherein each said task identifier is an
2 identifier of the object with which the task is associated.

1 15. (Original) A method as defined in claim 14 wherein each said dynamically
2 identified task is the garbage-collection task of performing, for a given object
3 associated with that task, processing that includes identifying in the given object
4 references to other objects and thereby identifying the tasks of performing similar
5 processing for those other objects.

1 16. (Original) A method as defined in claim 14 wherein each said task identifier is a
2 pointer to the object with which the task is associated
3

4 17. (Original) A method as defined in claim 12 wherein, when one said thread
5 identifies a task, the computer system pushes an identifier of that thread onto
6 only one of the queues of which that thread is an enqueueer.

1 18. (Original) A method as defined in claim 17 wherein each said dynamically
2 identified task is the garbage-collection task of performing, for a given object
3 associated with that task, processing that includes identifying in the given object
4 references to other objects and thereby identifying the tasks of performing similar
5 processing for those other objects.

1 19. (Original) A method as defined in claim 12 wherein identifiers of tasks
2 successively identified by a given thread are not in general pushed onto the
3 same queue.

1 20. (Original) A method as defined in claim 19 wherein each said dynamically
2 identified task is the garbage-collection task of performing, for a given object
3 associated with that task, processing that includes identifying in the given object
4 references to other objects and thereby identifying the tasks of performing similar
5 processing for those other objects.

1 21. (Original) A method as defined in claim 12 wherein a task queue is provided for
2 each ordered pair of the threads.

1 22. (Original) A method as defined in claim 21 wherein each said dynamically
2 identified task is the garbage-collection task of performing, for a given object
3 associated with that task, processing that includes identifying in the given object
4 references to other objects and thereby identifying the tasks of performing similar
5 processing for those other objects.

1 23. (Presently Amended) A storage medium containing instructions readable by a
2 computer system to configure the computer system to employ a plurality of
3 execution threads to perform dynamically identified tasks by:

- 4 A) providing a plurality of task queues, each of which is associated with a
5 different ordered pair of the threads, one thread of the ordered pair being
6 denominated the enqueueer of that queue and the other being
7 denominated the dequeueer thereof, wherein the execution threads operate
8 so that only an enqueueer of a task queue adds entries to that task queue
9 and only the dequeueer of a task queue removes entries from that task
10 queue;
- 11 B) when ~~one said~~ a thread identifies a task, pushing an identifier of ~~that the~~
12 task thus identified onto a set of at least one of the queues of which that
13 thread is an enqueueer; and
- 14 C) when ~~one said~~ a thread requires one of the dynamically identified tasks to
15 perform, causing that thread to perform a task identified by a task identifier
16 fetched by that thread from a task queue of which that thread is the
17 dequeueer.

1 24. (Original) A storage medium as defined in claim 23 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 25. (Original) A storage medium as defined in claim 23 wherein each said task
2 identifier is an identifier of the object with which the task is associated.

1 26. (Original) A storage medium as defined in claim 25 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the

4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 27. (Original) A storage medium as defined in claim 25 wherein each said task
2 identifier is a pointer to the object with which the task is associated.

1 28. (Original) A storage medium as defined in claim 23 wherein, when one said
2 thread identifies a task, the computer system pushes an identifier of that thread
3 onto only one of the queues of which that thread is an enqueueer.

1 29. (Original) A storage medium as defined in claim 28 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 30. (Original) A storage medium as defined in claim 23 wherein identifiers of tasks
2 successively identified by a given thread are not in general pushed onto the
3 same queue.

1 31. (Original) A storage medium as defined in claim 30 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 32. (Original) A storage medium as defined in claim 23 wherein a task queue is
2 provided for each ordered pair of the threads.

1 33. (Original) A storage medium as defined in claim 32 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 34. (Presently Amended) A computer signal representing a sequence of instructions
2 that, when executed by a computer system, cause the computer system to
3 employ a plurality of execution threads to perform dynamically identified tasks by:

4 A) provide a plurality of task queues, each of which is associated with a
5 different ordered pair of the threads, one thread of the ordered pair being
6 denominated the enqueueer of that queue and the other being
7 denominated the dequeueer thereof, wherein the execution threads operate
8 so that only an enqueueer of a task queue adds entries to that task queue
9 and only the dequeueer of a task queue removes entries from that task
10 queue;

11 B) ~~when one said~~ a thread identifies a task, pushes an identifier of that the
12 task thus identified onto a set of at least one of the queues of which that
13 thread is an enqueueer; and

14 C) ~~when one said~~ a thread requires one of the dynamically identified tasks to
15 perform, causes that thread to perform a task identified by a task identifier
16 fetched by that thread from a task queue of which that thread is the
17 dequeueer.

1 35. (Original) A computer signal as defined in claim 34 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the

4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 36. (Original) A computer signal as defined in claim 34 wherein each said task
2 identifier is an identifier of the object with which the task is associated.

1 37. (Original) A computer signal as defined in claim 36 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 38. (Original) A computer signal as defined in claim 36 wherein each said task
2 identifier is a pointer to the object with which the task is associated
3

4 39. (Original) A computer signal as defined in claim 34 wherein, when one said
5 thread identifies a task, the computer system pushes an identifier of that thread
6 onto only one of the queues of which that thread is an enqueueer.

1 40. (Original) A computer signal as defined in claim 39 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 41. (Original) A computer signal as defined in claim 34 wherein identifiers of tasks
2 successively identified by a given thread are not in general pushed onto the
3 same queue.

1 42. (Original) A computer signal as defined in claim 41 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 43. (Original) A computer signal as defined in claim 34 wherein a task queue is
2 provided for each ordered pair of the threads.

1 44. (Original) A computer signal as defined in claim 43 wherein each said
2 dynamically identified task is the garbage-collection task of performing, for a
3 given object associated with that task, processing that includes identifying in the
4 given object references to other objects and thereby identifying the tasks of
5 performing similar processing for those other objects.

1 45. (Presently Amended) A computer system that employs a plurality of execution
2 threads to perform tasks that the threads identify dynamically, the computer
3 system including:

4 A) means for providing a plurality of task queues, each of which is associated
5 with a different ordered pair of the threads, one thread of the ordered pair
6 being denominated the enqueuer of that queue and the other being
7 denominated the dequeuer thereof, wherein the execution threads operate
8 so that only an enqueuer of a task queue adds entries to that task queue
9 and only the dequeuer of a task queue removes entries from that task
10 queue;

- 11 B) means for, when ~~one said~~ a thread identifies a task, pushing an identifier
12 of ~~that~~ the task thus identified onto a set of at least one of the queues of
13 which that thread is an enqueueer; and
14 C) means for, when ~~one said~~ a thread requires one of the dynamically
15 identified tasks to perform, causing that thread to perform a task identified
16 by a task identifier fetched by that thread from a task queue of which that
17 thread is the dequeueer.